

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2002年 6月28日

出 願 番 号

Application Number:

特願2002-190890

[ST.10/C]:

[JP2002-190890]

出 願 人

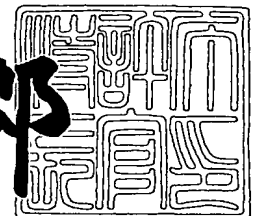
Applicant(s):

株式会社東芝

2002年10月11日

特 許 庁 長 官
Commissioner,
Japan Patent Office

太田 信一郎



出証番号 出証特2002-3079442

【書類名】 特許願

【整理番号】 A000202270

【提出日】 平成14年 6月28日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 13/00

【発明の名称】 クロック周波数の制御方法および電子機器

【請求項の数】 13

【発明者】

 【住所又は居所】 東京都青梅市末広町2丁目9番地 株式会社東芝青梅工場内

 【氏名】 山下 道生

【発明者】

 【住所又は居所】 東京都青梅市末広町2丁目9番地 株式会社東芝青梅工場内

 【氏名】 上床 克樹

【特許出願人】

 【識別番号】 000003078

 【氏名又は名称】 株式会社 東芝

【代理人】

 【識別番号】 100058479

 【弁理士】

 【氏名又は名称】 鈴江 武彦

 【電話番号】 03-3502-3181

【選任した代理人】

 【識別番号】 100084618

 【弁理士】

 【氏名又は名称】 村松 貞男

【選任した代理人】

 【識別番号】 100068814

【弁理士】

【氏名又は名称】 坪井 淳

【選任した代理人】

【識別番号】 100092196

【弁理士】

【氏名又は名称】 橋本 良郎

【選任した代理人】

【識別番号】 100091351

【弁理士】

【氏名又は名称】 河野 哲

【選任した代理人】

【識別番号】 100088683

【弁理士】

【氏名又は名称】 中村 誠

【選任した代理人】

【識別番号】 100070437

【弁理士】

【氏名又は名称】 河井 将次

【手数料の表示】

【予納台帳番号】 011567

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 クロック周波数の制御方法および電子機器

【特許請求の範囲】

【請求項 1】 プロセッサを備えた電子機器に適用されるクロック周波数の制御方法であって、

前記プロセッサの単位時間当りの実行命令数を取得し、

前記プロセッサの単位時間当りのクロック数を取得し、

前記クロック数に対する前記実行命令数の割合が所定値を超えているか否かを判定し、

前記判定の結果に応じて前記プロセッサのクロック周波数を制御することを特徴とするクロック周波数の制御方法。

【請求項 2】 前記実行命令数の取得、前記クロック数の取得、前記判定、及び前記クロック周波数の制御からなる一連の処理を、所定の時間間隔で繰り返し実行することを特徴とする請求項 1 記載のクロック周波数の制御方法。

【請求項 3】 プロセッサを備えた電子機器に適用されるクロック周波数の制御方法であって、

前記プロセッサの単位時間当りの実行命令数を取得し、

前記プロセッサの単位時間当りのクロック数を取得し、

前記クロック数に対する前記実行命令数の割合が所定値を超えていないかを判定し、

前記割合が所定値を超えていないと判定した場合には、前記プロセッサのクロック周波数を下げるように制御することを特徴とするクロック周波数の制御方法。

【請求項 4】 プロセッサを備えた電子機器に適用されるクロック周波数の制御方法であって、

前記プロセッサの単位時間当りの実行命令数を、所定の時間間隔で取得し、

前記プロセッサの単位時間当りのクロック数を取得し、

前記クロック数に対する、時間的に前後して取得される二つの実行命令数の差の割合が、所定値を超えているか否かを判定し、

前記判定の結果に応じて前記プロセッサのクロック周波数を制御することを特徴とするクロック周波数の制御方法。

【請求項 5】 プロセッサを備えた電子機器に適用されるクロック周波数の制御方法であって、

前記プロセッサの単位時間当りの実行命令数を、所定の時間間隔で取得し、
前記プロセッサの単位時間当りのクロック数を、所定の時間間隔で取得し、
時間的に前後して取得される個々のクロック数にそれぞれ所定の重み係数を乗じて互いに加算した値に対する、時間的に前後して取得される個々の実行命令数にそれぞれ所定の重み係数を乗じて互いに加算した値の割合が、所定値を超えているか否かを判定し、

前記判定の結果に応じて前記プロセッサのクロック周波数を制御することを特徴とするクロック周波数の制御方法。

【請求項 6】 前記判定の結果、前記割合が所定値を超えていると判定された場合には前記クロック周波数を上げ、前記割合が所定値を超えていないと判定された場合には前記クロック周波数を下げることの特徴とする請求項 1、4、5 のいずれかに記載のクロック周波数の制御方法。

【請求項 7】 前記所定の時間間隔を可変にすることを特徴とする請求項 2、4、5 のいずれかに記載のクロック周波数の制御方法。

【請求項 8】 前記所定値を可変にすることを特徴とする請求項 1、3、4、5 のいずれかに記載のクロック周波数の制御方法。

【請求項 9】 クロック信号を供給するクロック発振器と、
前記クロック発振器から供給されるクロック信号に基づいて内部クロックを生成するプロセッサと、

前記プロセッサが生成する内部クロックの単位時間当りのクロック数に対する、単位時間当りの実行命令数の割合に応じて、前記内部クロックの周波数を制御する制御手段と

を具備することを特徴とする電子機器。

【請求項 10】 前記制御手段は、前記クロック数に対する前記実行命令数の割合が所定値を超えているか否かを判定し、前記判定の結果に応じて前記プロ

セッサのクロック周波数を制御することを特徴とする請求項 9 記載の電子機器。

【請求項 1 1】 前記制御手段は、前記クロック数に対する、時間的に前後して取得される二つの実行命令数の差の割合が、所定値を超えているか否かを判定し、前記判定の結果に応じて前記プロセッサのクロック周波数を制御することを特徴とする請求項 9 記載の電子機器。

【請求項 1 2】 前記制御手段は、時間的に前後して取得される個々のクロック数にそれぞれ所定の重み係数を乗じて互いに加算した値に対する、時間的に前後して取得される個々の実行命令数にそれぞれ所定の重み係数を乗じて互いに加算した値の割合が、所定値を超えているか否かを判定し、前記判定の結果に応じて前記プロセッサのクロック周波数を制御することを特徴とする請求項 9 記載の電子機器。

【請求項 1 3】 前記制御手段は、前記割合が所定値を超えていると判定した場合には前記クロック周波数を上げ、前記割合が所定値を超えていないと判定した場合には前記クロック周波数を下げること特徴とする請求項 1 0 乃至 1 2 のいずれかに記載の電子機器。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は、プロセッサのクロック周波数の制御方法に関し、特に、プロセッサを備えた電子機器に適用される電力制御方法に関する。

【0 0 0 2】

【従来の技術】

従来、コンピュータなどの電子機器において省電力を行う方法の一つとして、プロセッサ（CPU など）の動作クロック周波数（もしくは動作速度）を低下させることによりプロセッサの消費電力を抑える方法がある。この方法では、プロセッサの使用率や、バッテリー容量、プロセッサの発熱状態などを監視し、それらの情報に基づいて省電力制御を行う。

【0 0 0 3】

例えば、プロセッサが I / O デバイスやメモリをアクセスするごとに所定の記

億領域にビジー状態などを示すフラグがセットされるものである場合、オペレーティングシステム（OS）（もしくはOSに付属するドライバソフトウェア）は、そのフラグのセット回数に基づいてプロセッサの負荷の度合いを取得し、これによりプロセッサの使用率を求めることができる。OSは、例えばプロセッサの使用率が低いと判断した場合には、省電力制御においてプロセッサのクロック周波数を低下させる。

【0004】

【発明が解決しようとする課題】

しかしながら、OSの側では、ストール処理に起因してプロセッサの実質的な使用率よりも高い値が認識されてしまう傾向がある。特に、I/Oアクセスの回数やキャッシュミスの回数が多い場合にはそのような現象が生じやすい。

【0005】

すなわち、プロセッサの使用率が実質的に低い状態であっても、OSはその使用率が高いものと判断してしまい、プロセッサのクロック周波数をなかなか低下させようとしめないという現象が起こり得る。そのような場合、プロセッサの消費電力を効率よく抑えることができないことになる。

【0006】

一方、特定のハードウェアを別途設けることによってプロセッサの実質的な使用率を検出できるようにすることも考えられる。しかしながら、そのような特定のハードウェアを新たに設けることは、コストの増大や実装スペースの拡張を招くことになるため、望ましいとは言えない。

【0007】

また、特開平11-353052号公報には、コンピュータのパフォーマンス指標と消費電力指標との比をエネルギー指標とし、このエネルギー指標に基づいてCPUの動作速度（動作周波数）を増減させることにより、省電力とパフォーマンスのバランスをとったパワー・マネージメント処理を実施するプロセッサ動作速度制御方法が開示されている。しかし、この文献の技術は、プロセッサの実質的な使用率に着目して省電力化を行うことを目指したものではない。

【0008】

本発明は上記実情に鑑みてなされたものであり、プロセッサの実質的な使用率に見合った消費電力を制御する方法および電子機器を提供することを目的とする。

【 0 0 0 9 】

【課題を解決するための手段】

本発明に係るクロック周波数の制御方法は、プロセッサを備えた電子機器に適用されるクロック周波数の制御方法であって、前記プロセッサの単位時間当りの実行命令数を取得し、前記プロセッサの単位時間当りのクロック数を取得し、前記クロック数に対する前記実行命令数の割合が所定値を超えているか否かを判定し、前記判定の結果に応じて前記プロセッサのクロック周波数を制御することを特徴とする。

【 0 0 1 0 】

また、本発明に係るクロック周波数の制御方法は、プロセッサを備えた電子機器に適用されるクロック周波数の制御方法であって、前記プロセッサの単位時間当りの実行命令数を取得し、前記プロセッサの単位時間当りのクロック数を取得し、前記クロック数に対する前記実行命令数の割合が所定値を超えていないかを判定し、前記割合が所定値を超えていないと判定した場合には、前記プロセッサのクロック周波数を下げるように制御することを特徴とする。

【 0 0 1 1 】

また、本発明に係るクロック周波数の制御方法は、プロセッサを備えた電子機器に適用されるクロック周波数の制御方法であって、前記プロセッサの単位時間当りの実行命令数を、所定の時間間隔で取得し、前記プロセッサの単位時間当りのクロック数を取得し、前記クロック数に対する、時間的に前後して取得される二つの実行命令数の差の割合が、所定値を超えているか否かを判定し、前記判定の結果に応じて前記プロセッサのクロック周波数を制御することを特徴とする。

【 0 0 1 2 】

また、本発明に係るクロック周波数の制御方法は、プロセッサを備えた電子機器に適用されるクロック周波数の制御方法であって、前記プロセッサの単位時間当りの実行命令数を、所定の時間間隔で取得し、前記プロセッサの単位時間当り

のクロック数を、所定の時間間隔で取得し、時間的に前後して取得される個々のクロック数にそれぞれ所定の重み係数を乗じて互いに加算した値に対する、時間的に前後して取得される個々の実行命令数にそれぞれ所定の重み係数を乗じて互いに加算した値の割合が、所定値を超えているか否かを判定し、前記判定の結果に応じて前記プロセッサのクロック周波数を制御することを特徴とする。

【 0 0 1 3 】

また、本発明に係る電子機器は、クロック信号を供給するクロック発振器と、前記クロック発振器から供給されるクロック信号に基づいて内部クロックを生成するプロセッサと、前記プロセッサが生成する内部クロックの単位時間当りのクロック数に対する、単位時間当りの実行命令数の割合に応じて、前記内部クロックの周波数を制御する制御手段とを具備することを特徴とする。

【 0 0 1 4 】

【発明の実施の形態】

以下、図面を参照して本発明の実施形態を説明する。

【 0 0 1 5 】

図 1 は、本発明の一実施形態に係る電子機器の要部構成を示すブロック図である。

【 0 0 1 6 】

本実施形態に係る電子機器は、例えばパーソナルコンピュータなどの情報機器であり、主メモリ 1、CPU (Central Processing Unit) 2、クロック発振器 3、I/O (Input/output) デバイス 4、記録媒体 5などを備えている。

【 0 0 1 7 】

主メモリ 1は、CPU 2のワークエリアとして提供され、CPU 2により処理されるOS (Operating System) 11、アプリケーション 12、パワー・マネジメント・ドライバ 13 (OS 11に含まれる)などを保持している。

【 0 0 1 8 】

OS 11は、アプリケーション 12の動作を制御すると共に、パワー・マネジメント・ドライバ 13の動作をも制御する。アプリケーション 12は、OS 11の管理下で動作するソフトウェアである。また、パワー・マネジメント・ド

ライバ 1 3 は、特定のハードウェアに対応した制御を行うソフトウェアドライバの一種であり、特にパワー・マネージメントを行うためのドライバである。

【 0 0 1 9 】

パワー・マネージメント・ドライバ 1 3 には、制御部 1 4 及びタイマ 1 5 が備えられる。制御部 1 4 は、本実施形態における省電力を実現するためのソフトウェアに相当するものであり、CPU 2 の単位時間当りのクロック数（以下、単にクロック数と称す場合がある）に対する、単位時間当りの実行命令数（以下、単に実行命令数と称す場合がある）の割合に応じて、CPU 2 で生成される内部クロックの周波数を制御する制御機能を有する。上記クロック数および実行命令数については、CPU 2 から知得することが可能である。上記制御機能の詳細については後で述べる。また、タイマ 1 5 は、制御部 1 4 によって使用されるものであり、後述する Wait 時間などをカウントする際に使用される。

【 0 0 2 0 】

CPU 2 は、電子機器全体の制御を司るものであり、主メモリ 1 2 をワークエリアとして使用し、各種ソフトウェアを実行する。この CPU 2 には、実行命令数を保持するレジスタ 2 1 が内蔵されている。レジスタ 2 1 に保持される実行命令数は、CPU 2 の処理状況によって変化する。実行命令数は、CPU 2 の外部から参照することが可能である。

【 0 0 2 1 】

また、CPU 2 には、クロック処理回路 2 2 が内蔵されている。このクロック処理回路 2 2 は、クロック発振器 3 から供給されるクロック信号に基づき、CPU 2 で使用する内部クロックを発生するものである。また、このクロック処理回路 2 2 は、制御部 1 4 からの指示に従って、内部クロックの周波数を変更する。

【 0 0 2 2 】

クロック発振器 3 は、基本クロック信号を生成して CPU 2 （の中のクロック処理回路 2 2 ）へ供給するものである。また、I/O デバイス 4 は、CPU 2 との間でデータの入出力を行うデバイスである。

【 0 0 2 3 】

記録媒体 5 は、HDD (Hard Disk Drive) などに相当するものであり、本電

子機器がパワーオフされる際には、上記OS 11、アプリケーション12、パワー・マネージメント・ドライバ13などの各種ソフトウェアやデータが保存される。

【0024】

図2は、図1における制御部14の機能構成の一例を示すブロック図である。

【0025】

制御部14は、実行命令数／クロック数取得機能141、判定機能142、閾値テーブル143、クロック周波数変更指示機能144、Wait時間設定機能145を備えている。

【0026】

実行命令数／クロック数取得機能141は、Wait時間設定機能145によって所定の時間間隔で起動され、CPU2のレジスタ21にアクセスすることによって実行命令数を取得すると共に、クロック処理回路にアクセスすることによってクロック数を取得するものである。これらの情報は判定機能142によって使用される。

【0027】

判定機能142は、実行命令数／クロック数取得機能141によって起動され、実行命令数／クロック数取得機能141によって取得された実行命令数およびクロック数を用いて所定の演算（後述）を行い、その演算結果が閾値（所定値）を超えているか否かに応じてCPU2のクロック周波数を上げるべきか下げるべきか（あるいは不変とすべきか）を判定するものである。

【0028】

上記判定機能142で使用される演算の例としては、以下のものが挙げられる。

【0029】

（演算例1）

演算「 I / I_0 」（但し、 I は単位時間当りの実行命令数、 I_0 は単位時間当りのクロック数）を実行する。

【0030】

(演算例 2)

演算「 $(I_n - I_{n-1}) / I_0$ 」(但し、 I_n は n 番目に取得された単位時間当りの実行命令数、 I_0 は単位時間当りのクロック数)、もしくは演算「 $(I_{n-1} - I_n) / I_0$ 」を実行する。

【0031】

(演算例 3)

演算「 $(I_n * \alpha_n + I_{n-1} * \alpha_{n-1} + \dots) / (I_{0n} * \alpha_n + I_{0(n-1)} * \alpha_{n-1} + \dots)$ 」(但し、 I_n は n 番目に取得された単位時間当りの実行命令数、 I_{0n} は n 番目に取得された単位時間当りのクロック数、 α_n は n 番目に取得された情報に適用される重み係数)を実行する。この場合、例えば、 α_n よりも α_{n-1} の方を小さい値にし、 α_{n-1} よりも α_{n-2} の方を小さい値にし、…とすることにより、過去に取得された情報よりも最新の情報に重きをおいた演算を実行させることが可能である。なお、重み係数 α_n 、 α_{n-1} 、…は、予め所定の記憶領域に記憶しておいた基準の重み係数 α に基づき、アルゴリズム等を用いて適宜算出される。

【0032】

上記判定機能142は、上記判定に用いる閾値(所定値)を、閾値テーブル143を参照して決定する。閾値テーブル143には、図3に示されるように、各種モード(CPU2の動作モード)の種類に応じて使用されるべき閾値 S の具体的な値が記述されている。図3の例では、CPU2が省電力モードのときは閾値 S を0.9とし、CPU2が標準モードのときは閾値 S を0.5とし、CPU2が高速モードのときは閾値 S を0.1とすることを示している。すなわち、判定機能142は、CPU2の動作モードに対応する閾値(所定値)を、閾値テーブル143を参照することにより決定する。なお、CPU2の動作モード(省電力モード・標準モード・高速モードなど)の情報は、パワー・マネージメント・ドライバ13から取得できるようになっている。

【0033】

クロック周波数変更指示機能144は、判定機能142によって起動され、判定機能142での演算結果が閾値(所定値)を超えているか否かに応じてCPU2のクロック周波数を所定分だけ上げ/下げさせる指示をクロック処理回路22

へ送る。この場合のクロック周波数の上げ／下げは、ユーザ側から見てパフォーマンスの変化が殆ど感じられない程度に行う。なお、CPU 2 の現在のクロック周波数は、CPU 2 から取得できるようになっている。

【 0 0 3 4 】

本実施形態ではクロック処理回路 2 2 に対して CPU 2 のクロック周波数の変更指示を行う構成を示しているが、代わりに、クロック発振器 3 に対して生成されるクロック周波数の変更指示を行うように構成してもよい。

【 0 0 3 5 】

Wait 時間設定機能 1 4 5 は、クロック周波数変更指示機能 1 4 4 によって起動され、予め設定されている待ち時間 (Wait 時間) を経過した後に、実行命令数／クロック数取得機能 1 4 1 を起動する。なお、ここで設定されている Wait 時間は、必要に応じて変更することが可能である。例えば、現在設定されている CPU 2 の動作モードの種類に相応しい Wait 時間となるように設定変更を行うようにしてもよい。

【 0 0 3 6 】

次に、図 1 ～ 2 を参照すると共に、図 4 ～ 図 6 のフローチャートを参照して、本実施形態における 3 つの動作例を説明する。

【 0 0 3 7 】

(第 1 の動作例)

図 4 は、前述の演算例 1 に対応した動作を示している。

【 0 0 3 8 】

電子機器の起動時においては、判定機能 1 4 2 は、CPU 2 の動作モードの種類を認識し、その動作モードに対応する閾値 S を、閾値テーブル 1 4 3 を参照することによって予め取得する (ステップ A 1)。尚、閾値 S は、動作モード変更時にそのモードに対応する値に変わる。

【 0 0 3 9 】

制御部 1 4 の実行命令数／クロック数取得機能 1.4 1 により、単位時間当りの実行命令数 I 、単位時間当りのクロック数 I_0 が取得され、これらの情報が判定機能 1 4 2 に送られる (ステップ A 2、A 3)。

【 0 0 4 0 】

次に、判定機能 1 4 2 は、演算「 I / I_0 」（但し、 I は単位時間当りの実行命令数、 I_0 は単位時間当りのクロック数）を実行し、その演算結果が閾値 S を超えているか否かを判定する（ステップ A 4）。

【 0 0 4 1 】

判定機能 1 4 2 により演算結果が閾値 S を超えていると判定された場合（ステップ A 4 の YES）、クロック周波数変更指示機能 1 4 4 は、CPU 2 のクロック周波数が最高になっていなければ（ステップ A 5 の NO）、そのクロック周波数を所定分だけ上げるための制御を行う（ステップ A 6）。なお、CPU 2 のクロック周波数が最高になっていれば（ステップ A 5 の YES）、クロック周波数の変更は行わない。

【 0 0 4 2 】

一方、判定機能 1 4 2 により演算結果が閾値 S を超えていないと判定された場合（ステップ A 4 の NO）、クロック周波数変更指示機能 1 4 4 は、CPU 2 のクロック周波数が最低になっていなければ（ステップ A 7 の NO）、そのクロック周波数を所定分だけ下げるための制御を行う（ステップ A 8）。なお、CPU 2 のクロック周波数が最低になっていれば（ステップ A 7 の YES）、クロック周波数の変更は行わない。

【 0 0 4 3 】

この後、Wait 時間設定機能 1 4 5 は、Wait 時間の設定変更を行うべきか否かを判定する（ステップ A 9）。設定変更を行う場合、Wait 時間設定機能 1 4 5 は Wait 時間の設定変更を行い（ステップ A 1 0）、変更後の Wait 時間の経過を待ち（ステップ A 1 1）、Wait 時間の経過後に実行命令数／クロック数取得機能 1 4 1 を起動する。一方、設定変更を行わない場合は、Wait 時間設定機能 1 4 5 は、既に設定されている Wait 時間の経過を待ち（ステップ A 1 1）、Wait 時間の経過後に実行命令数／クロック数取得機能 1 4 1 を起動する。

【 0 0 4 4 】

このように第 1 の動作例では、クロック数に対する実行命令数の割合が所定値を超えているか否かを判定し、その判定結果に応じて CPU のクロック周波数を

制御することにより、CPUの実質的な使用率に見合った省電力化を行っている。

【0045】

(第2の動作例)

図5は、前述の演算例2に対応した動作を示している。

【0046】

電子機器の起動時においては、判定機能142は、CPU2の動作モードの種類を認識し、その動作モードに対応する閾値Sを、閾値テーブル143を参照することによって予め取得する(ステップB1)。

【0047】

次に、これから行う実行命令数/クロック数の取得が何番目であることを示すカウント値nを0とし、これを所定の記憶領域に記憶しておく(ステップB2)。

【0048】

実行命令数/クロック数の取得を開始するため、現在のカウント値に1を加算する(ステップB3)。

【0049】

制御部14の実行命令数/クロック数取得機能141により、単位時間当りの実行命令数I、単位時間当りのクロック数 I_0 が取得され、これらの情報が判定機能142に送られる(ステップB4、B5)。

【0050】

次に、判定機能142は、演算「 $(I_n - I_{n-1}) / I_0$ 」(但し、 I_n はn番目に取得された単位時間当りの実行命令数、 I_0 は単位時間当りのクロック数)を実行し、その演算結果が閾値Sを超えているか否かを判定する(ステップB6)。

【0051】

判定機能142により演算結果が閾値Sを超えていると判定された場合(ステップB6のYES)、クロック周波数変更指示機能144は、CPU2のクロック周波数が最高になっていなければ(ステップB7のNO)、そのクロック周波数を所定分だけ上げるための制御を行う(ステップB8)。なお、CPU2のク

ロック周波数が最高になっていれば（ステップ B 7 の Y E S）、クロック周波数の変更は行わない。

【 0 0 5 2 】

一方、判定機能 1 4 2 により演算結果が閾値 S を超えていないと判定された場合（ステップ B 6 の N O）、判定機能 1 4 2 は演算「 $(I_{n-1} - I_n) / I_0$ 」を実行し、その演算結果が閾値 S を超えているか否かを判定する（ステップ B 9）。ここで、判定機能 1 4 2 により演算結果が閾値 S を超えていると判定された場合（ステップ B 9 の Y E S）、クロック周波数変更指示機能 1 4 4 は、CPU 2 のクロック周波数が最低になっていなければ（ステップ B 1 0 の N O）、そのクロック周波数を所定分だけ下げるための制御を行う（ステップ B 1 1）。なお、CPU 2 のクロック周波数が最低になっていれば（ステップ B 1 0 の Y E S）、クロック周波数の変更は行わない。

【 0 0 5 3 】

この後、Wait 時間設定機能 1 4 5 は、Wait 時間の設定変更を行うべきか否かを判定する（ステップ B 1 2）。設定変更を行う場合、Wait 時間設定機能 1 4 5 は Wait 時間の設定変更を行い（ステップ B 1 3）、変更後の Wait 時間の経過を待ち（ステップ B 1 4）、Wait 時間の経過後に実行命令数／クロック数取得機能 1 4 1 を起動する。一方、設定変更を行わない場合は、Wait 時間設定機能 1 4 5 は、既に設定されている Wait 時間の経過を待ち（ステップ B 1 4）、Wait 時間の経過後に実行命令数／クロック数取得機能 1 4 1 を起動する。

【 0 0 5 4 】

このように第 2 の動作例では、前記クロック数に対する、時間的に前後して取得される二つの実行命令数の差の割合が、所定値を超えているか否かを判定し、その判定結果に応じて CPU のクロック周波数を制御することにより、CPU の実質的な使用率に見合った省電力化を行っている。

【 0 0 5 5 】

（第 3 の動作例）

図 6 は、前述の演算例 3 に対応した動作を示している。

【 0 0 5 6 】

電子機器の起動時においては、判定機能 1 4 2 は、CPU 2 の動作モードの種類を認識し、その動作モードに対応する閾値 S を、閾値テーブル 1 4 3 を参照することによって予め取得する。同時に、所定の記憶領域に記憶しておいた重み係数 α を予め取得する（ステップ C 1）。

【0 0 5 7】

次に、これから行う実行命令数／クロック数の取得が何番目であるかを示すカウント値 n を 0 とし、これを所定の記憶領域に記憶しておく（ステップ C 2）。

【0 0 5 8】

実行命令数／クロック数の取得を開始するため、現在のカウント値に 1 を加算する（ステップ C 3）。

【0 0 5 9】

制御部 1 4 の実行命令数／クロック数取得機能 1 4 1 により、単位時間当りの実行命令数 I 、単位時間当りのクロック数 I_0 が取得され、これらの情報が判定機能 1 4 2 に送られる（ステップ C 4、C 5）。

【0 0 6 0】

次に、判定機能 1 4 2 は、演算「 $(I_n * \alpha_n + I_{n-1} * \alpha_{n-1} + \dots) / (I_{0n} * \alpha_n + I_{0(n-1)} * \alpha_{n-1} + \dots)$ 」（但し、 I_n は n 番目に取得された単位時間当りの実行命令数、 I_{0n} は n 番目に取得された単位時間当りのクロック数、 α_n は n 番目に取得された情報に適用される重み係数）を実行し、その演算結果が閾値 S を超えているか否かを判定する（ステップ C 6）。

【0 0 6 1】

なお、上記演算において、分子の数式と分母の数式に使用すべき項数を所定の数に限定するようにしてもよい。この場合、演算を実行する毎に、最も古く取得された情報に関わる項を演算から適用除外（削除）すればよい。このようにすると、記憶領域に保持しておくべき情報の容量を低減することが可能となる。

【0 0 6 2】

判定機能 1 4 2 により演算結果が閾値 S を超えていると判定された場合（ステップ C 6 の YES）、クロック周波数変更指示機能 1 4 4 は、CPU 2 のクロック周波数が最高になっていなければ（ステップ C 7 の NO）、そのクロック周波

数を所定分だけ上げるための制御を行う（ステップC 8）。なお、CPU 2のクロック周波数が最高になっていれば（ステップC 7のYES）、クロック周波数の変更は行わない。

【0063】

一方、判定機能142により演算結果が閾値Sを超えていないと判定された場合（ステップC 6のNO）、クロック周波数変更指示機能144は、CPU 2のクロック周波数が最低になっていなければ（ステップC 9のNO）、そのクロック周波数を所定分だけ下げるための制御を行う（ステップC 10）。なお、CPU 2のクロック周波数が最低になっていれば（ステップC 9のYES）、クロック周波数の変更は行わない。

【0064】

この後、Wait時間設定機能145は、Wait時間の設定変更を行うべきか否かを判定する（ステップC 11）。設定変更を行う場合、Wait時間設定機能145はWait時間の設定変更を行い（ステップC 12）、変更後のWait時間の経過を待ち（ステップC 13）、Wait時間の経過後に実行命令数／クロック数取得機能141を起動する。一方、設定変更を行わない場合は、Wait時間設定機能145は、既に設定されているWait時間の経過を待ち（ステップC 13）、Wait時間の経過後に実行命令数／クロック数取得機能141を起動する。

【0065】

このように第3の動作例では、時間的に前後して取得される個々のクロック数にそれぞれ所定の重み係数を乗じて互いに加算した値に対する、時間的に前後して取得される個々の実行命令数にそれぞれ所定の重み係数を乗じて互いに加算した値の割合が、所定値を超えているか否かを判定し、その判定結果に応じてCPUのクロック周波数を制御することにより、CPUの実質的な使用率に見合った省電力化を行っている。

【0066】

以上説明したように本実施形態によれば、CPU 2の単位時間当りのクロック数に対する、単位時間当りの実行命令数の割合に応じて、CPU 2で生成される内部クロックの周波数を制御する制御機能を設けることにより、CPU 2の実質

的な使用率に見合った消費電力の制御を行うことができる。

【 0 0 6 7 】

なお、本発明は、上述した実施形態に限定されるものではなく、その要旨を逸脱しない範囲内で種々変形して実施することが可能である。例えば制御部の機能構成は図 2 に示したものに限定されるものではなく、同じ動作を実現できるのであれば機能構成を変形しても構わない。

【 0 0 6 8 】

【発明の効果】

以上詳記したように本発明によれば、プロセッサの実質的な使用率に見合った消費電力を制御することができる。

【図面の簡単な説明】

【図 1】

本発明の一実施形態に係る電子機器の要部構成を示すブロック図。

【図 2】

図 1 における制御部の機能構成の一例を示すブロック図。

【図 3】

図 2 における閾値テーブルの内容を説明するための図。

【図 4】

同実施形態における第 1 の動作例を示すフローチャート。

【図 5】

同実施形態における第 2 の動作例を示すフローチャート。

【図 6】

同実施形態における第 3 の動作例を示すフローチャート。

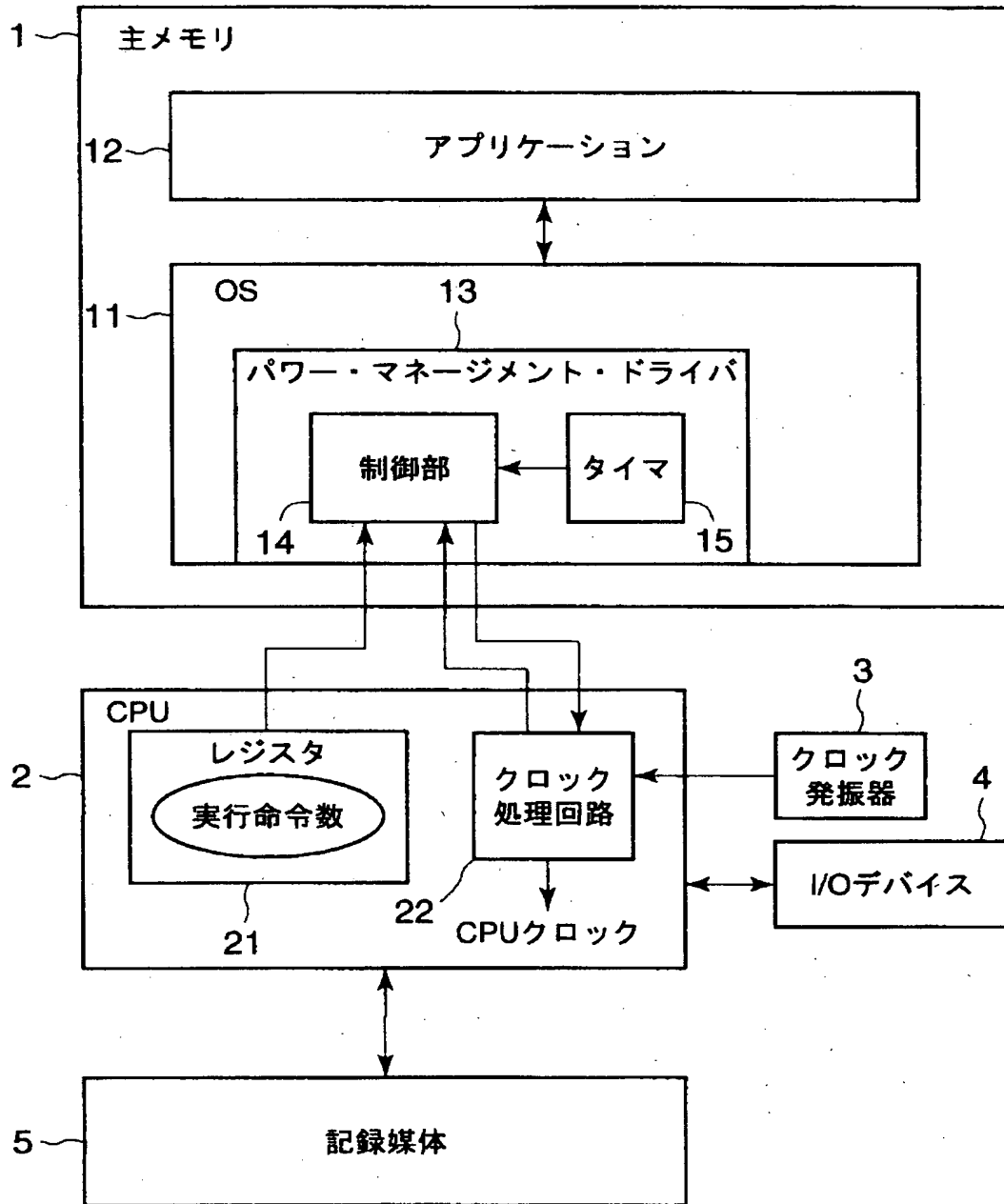
【符号の説明】

- 1 …主メモリ
- 2 …CPU
- 3 …クロック発振器
- 4 …I/Oデバイス
- 5 …記録媒体

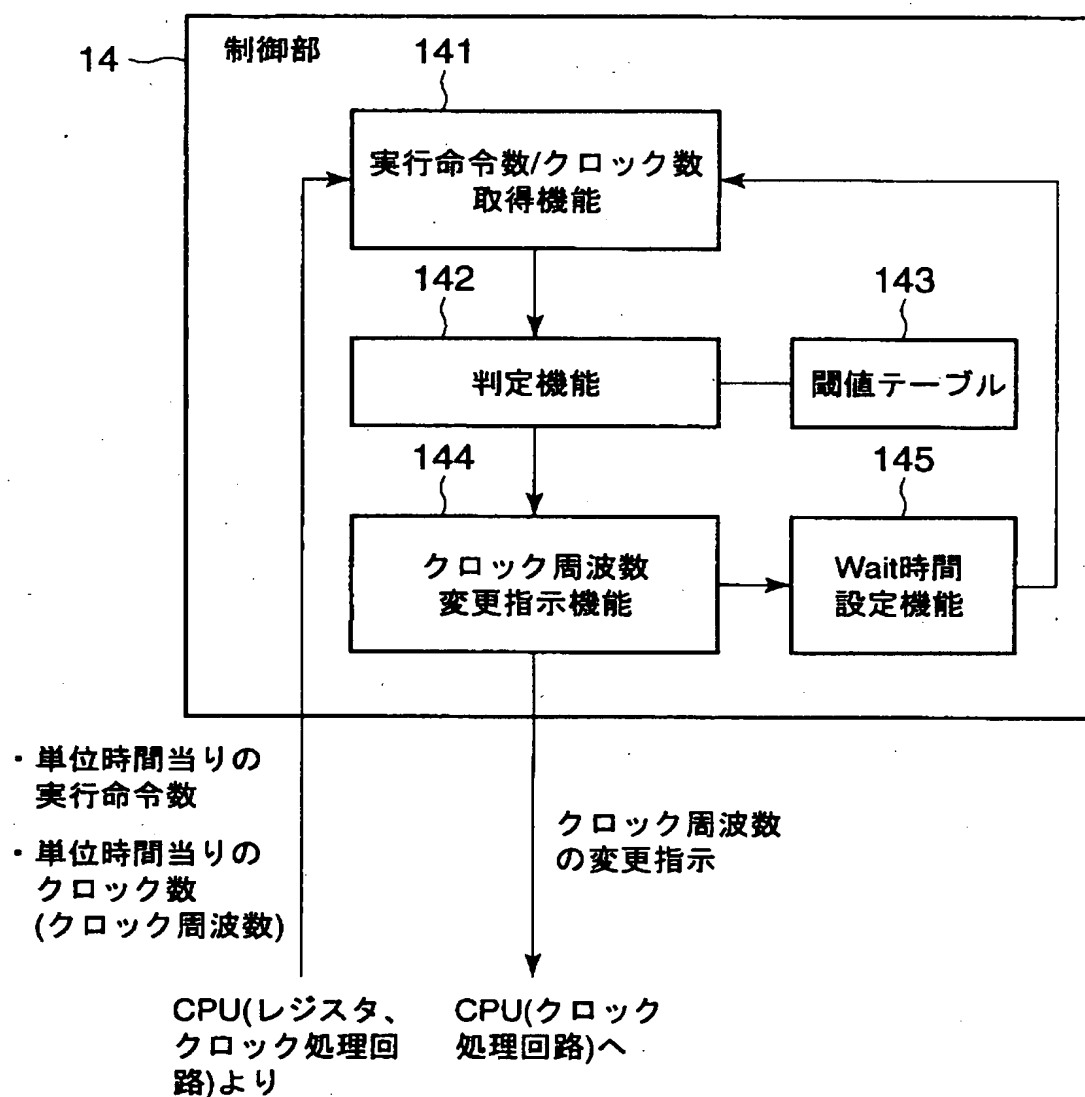
- 1 1 … OS
- 1 2 … アプリケーション
- 1 3 … パワー・マネージメント・ドライバ
- 1 4 … 制御部
- 1 5 … タイマ
- 2 1 … レジスタ
- 2 2 … クロック処理回路
- 1 4 1 … 実行命令数／クロック数取得機能
- 1 4 2 … 判定機能
- 1 4 3 … 閾値テーブル
- 1 4 4 … クロック周波数変更指示機能
- 1 4 5 … Wait時間設定機能

【書類名】 図面

【図1】



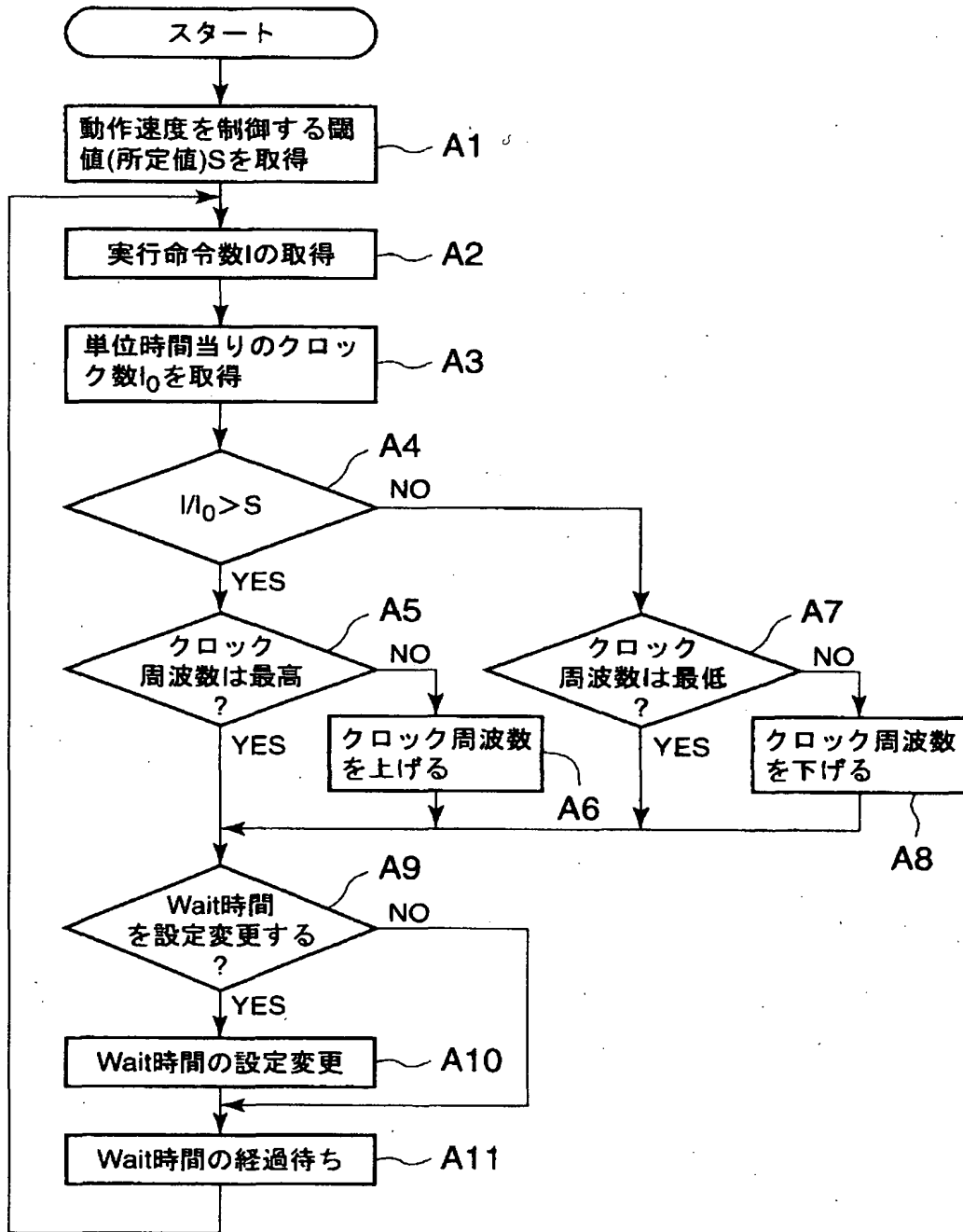
【図 2】



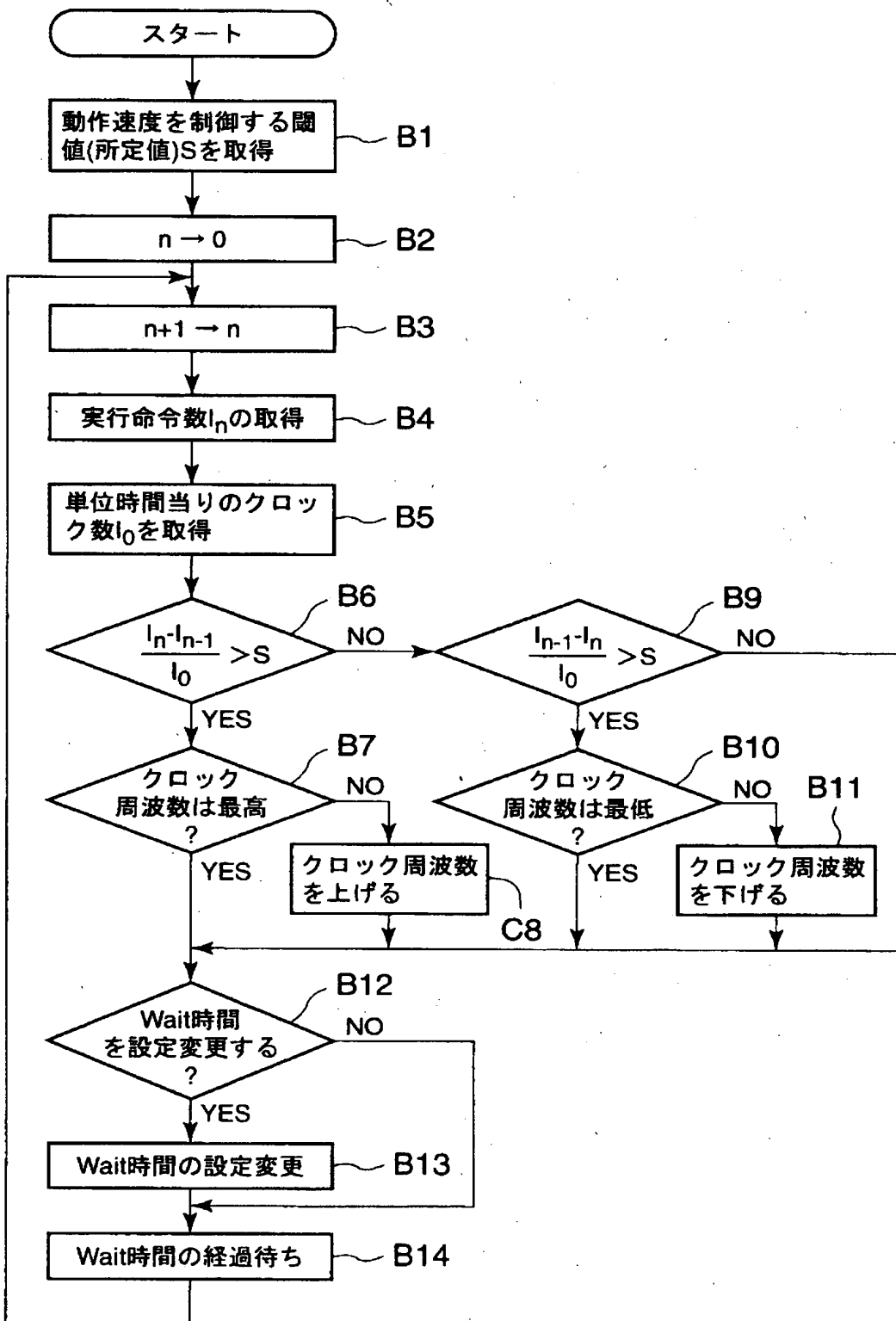
【図 3】

| S | モードの種類 |
|-----|--------|
| 0.9 | 省電力モード |
| 0.5 | 標準モード |
| 0.1 | 高速モード |

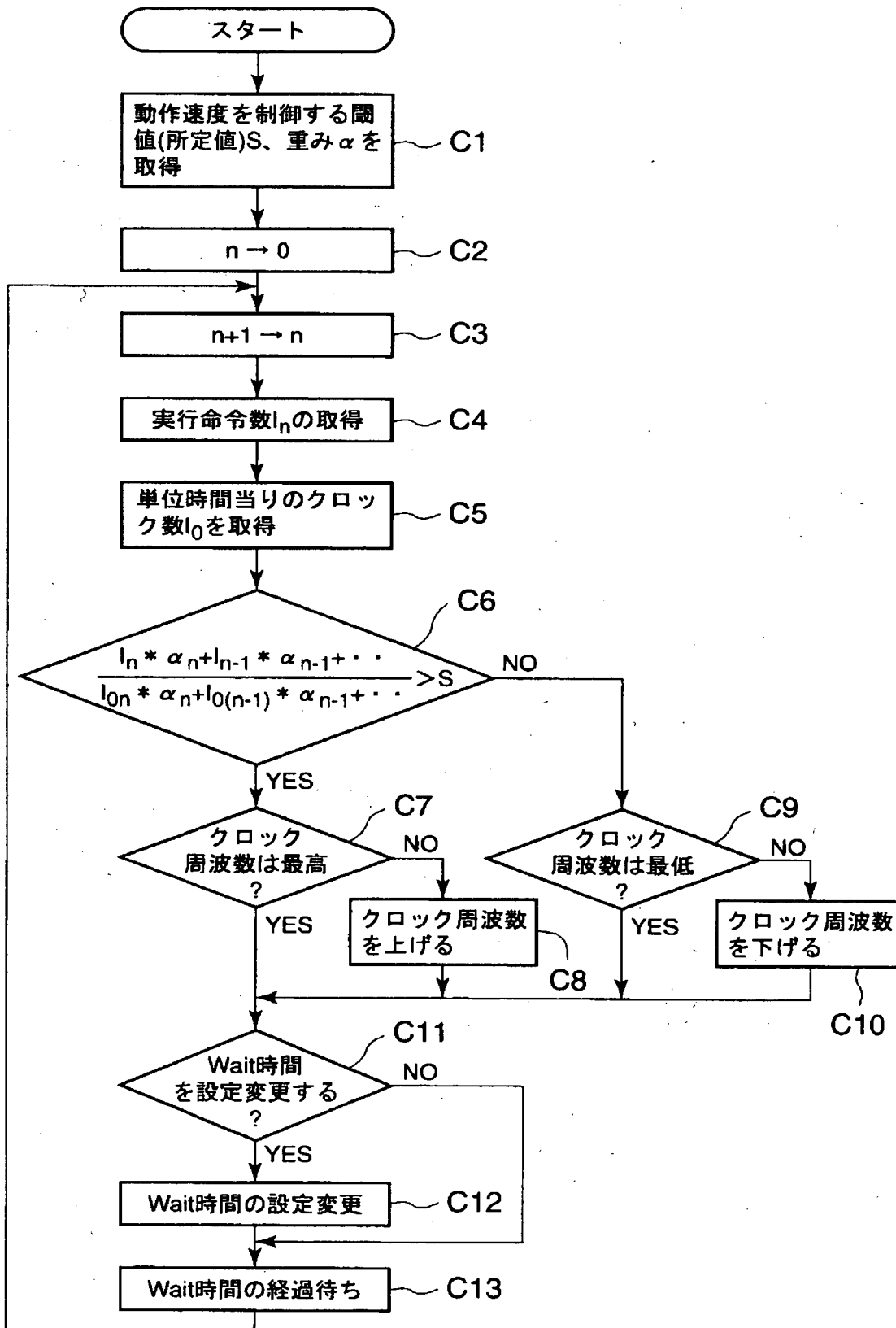
【図 4】



【図 5】



【図 6】



【書類名】 要約書

【要約】

【課題】 プロセッサの実質的な使用率に見合った消費電力を制御することができるようにする。

【解決手段】 CPU 2 には、単位時間当りの実行命令数を保持するレジスタ 2 1 が内蔵されている。レジスタ 2 1 に保持される実行命令数は、CPU 2 の処理状況によって変化する。また、CPU 2 には、クロック処理回路 2 2 が内蔵されている。このクロック処理回路 2 2 は、クロック発振器 3 から供給されるクロック信号に基づき、CPU 2 で使用する内部クロックを発生する。制御部 1 4 は、省電力を実現するためのソフトウェアに相当するものであり、CPU 2 の単位時間当りのクロック数に対する、単位時間当りの実行命令数の割合に応じて、CPU 2 の内部クロックの周波数を制御する制御機能を有する。クロック処理回路 2 2 は、制御部 1 4 からの指示に従って、内部クロックの周波数を変更する。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号 [000003078]

| | |
|----------|----------------|
| 1. 変更年月日 | 2001年 7月 2日 |
| [変更理由] | 住所変更 |
| 住 所 | 東京都港区芝浦一丁目1番1号 |
| 氏 名 | 株式会社東芝 |